

# **RAPPORT de Programmation et** **Conception Orientée Objet** **Système de Clavardage**



Valentin GUERBETTE & Lauric MARTHRIN-JOHN  
Ingénieur INSA en 4ème Année Informatique et Réseaux  
UF POO et COO - Année 2021/2022

# TABLES DES MATIÈRES

<b>TABLES DES MATIÈRES</b>	<b>2</b>
<b>INTRODUCTION</b>	<b>3</b>
Conception et diagrammes réalisés	3
DIAGRAMME DE SÉQUENCES	4
DIAGRAMME DE CLASSES	6
DIAGRAMME DE CAS D'UTILISATION	6
Architecture du système et choix technologiques	7
<b>Choix techniques</b>	<b>7</b>
1. Choix sur le stockage de données de conversation entres utilisateurs	7
2. Choix sur la signalisation entre les utilisateurs du Chat System	7
<b>Choix conceptuels</b>	<b>8</b>
Choix de création et de modification du compte utilisateur	8
Répartition du travail	8
Procédures d'évaluation et de tests	9
Procédure d'installation et de déploiement	10
Manuel d'utilisation simplifié	11
Mode : FRONT END	11
Mode : BACK END	11
<b>CONCLUSION</b>	<b>12</b>
<b>BIBLIOGRAPHIE</b>	<b>12</b>

## INTRODUCTION

Ce rapport présente tous les éléments techniques, qui ont été réalisés pour l'élaboration du produit, et nous a permis de d'implémenter et de concevoir une application de système de clavardage.

Nous avons eu comme projet de réaliser une application de discussion pour une entreprise de renommée nationale et internationale tel que THALES ALENIA SPACE de façon fictive.

Grâce à cette application, les utilisateurs pourront envoyer et recevoir des messages, des images, ou autre type de fichiers.

Nous possédons un répertoire sur GitHub avec tous les codes sources pour réaliser cette application. Il y a les fichiers JAVA, les ressources et les tests et le dossier jar.

- lien du github : <https://github.com/scanim-valentin/decentralized-chat-system>

Pour ce faire, nous avons reçu le cahier des charges dans lequel le lien est url est en fin du rapport et un libre choix de commencer par où l'on souhaiterait.

Il nous a été demandé de fournir pour ce rapport :

1. Conception et diagrammes réalisés
2. Architecture du système et choix technologiques
3. Procédures d'évaluation et de tests
4. Procédure d'installation et de déploiement
5. Manuel d'utilisation simplifié

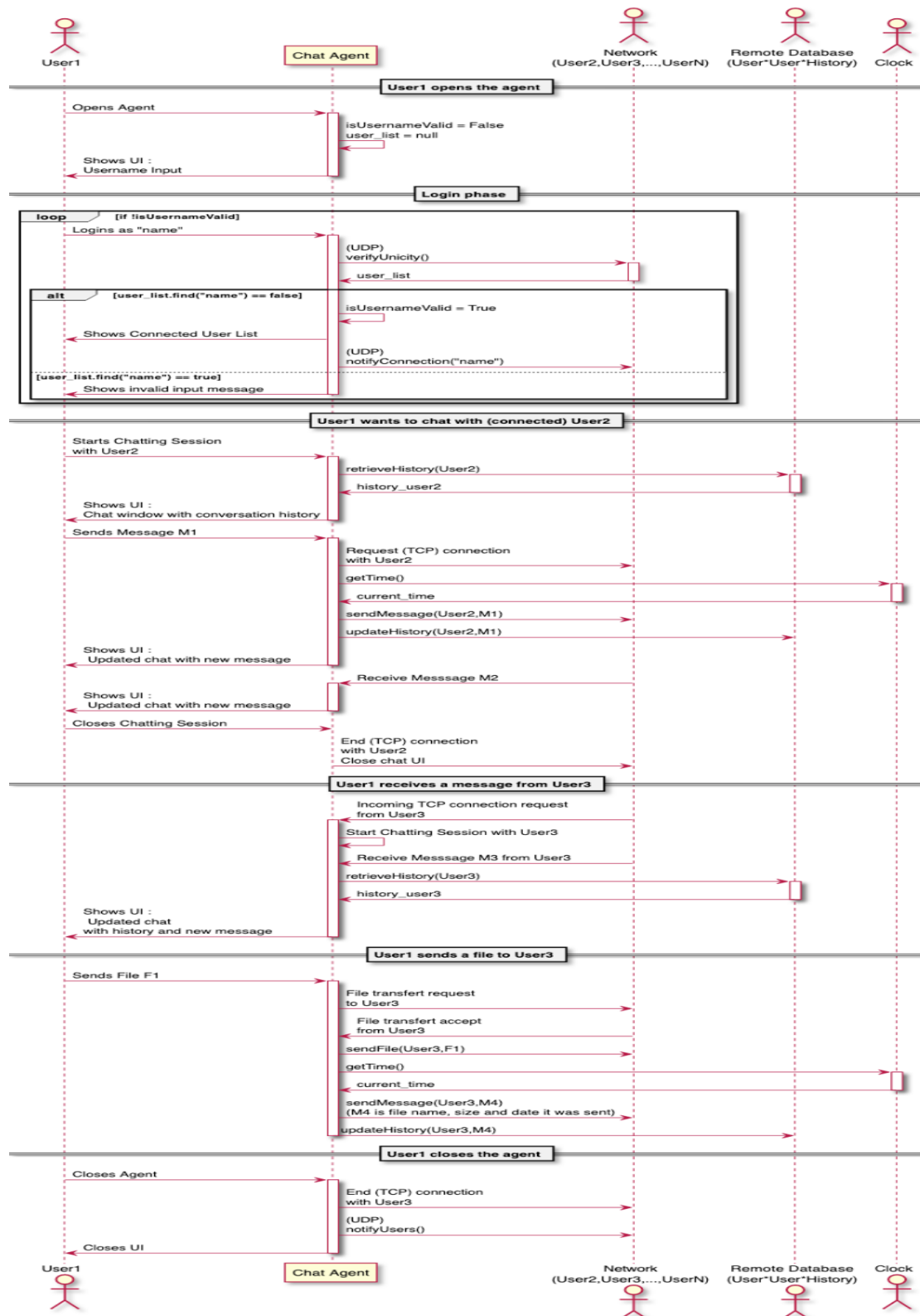
## Conception et diagrammes réalisés

Ce rapport présente en profondeur l'ensemble des diagrammes UML qui nous ont permis de concevoir le système de clavardage.

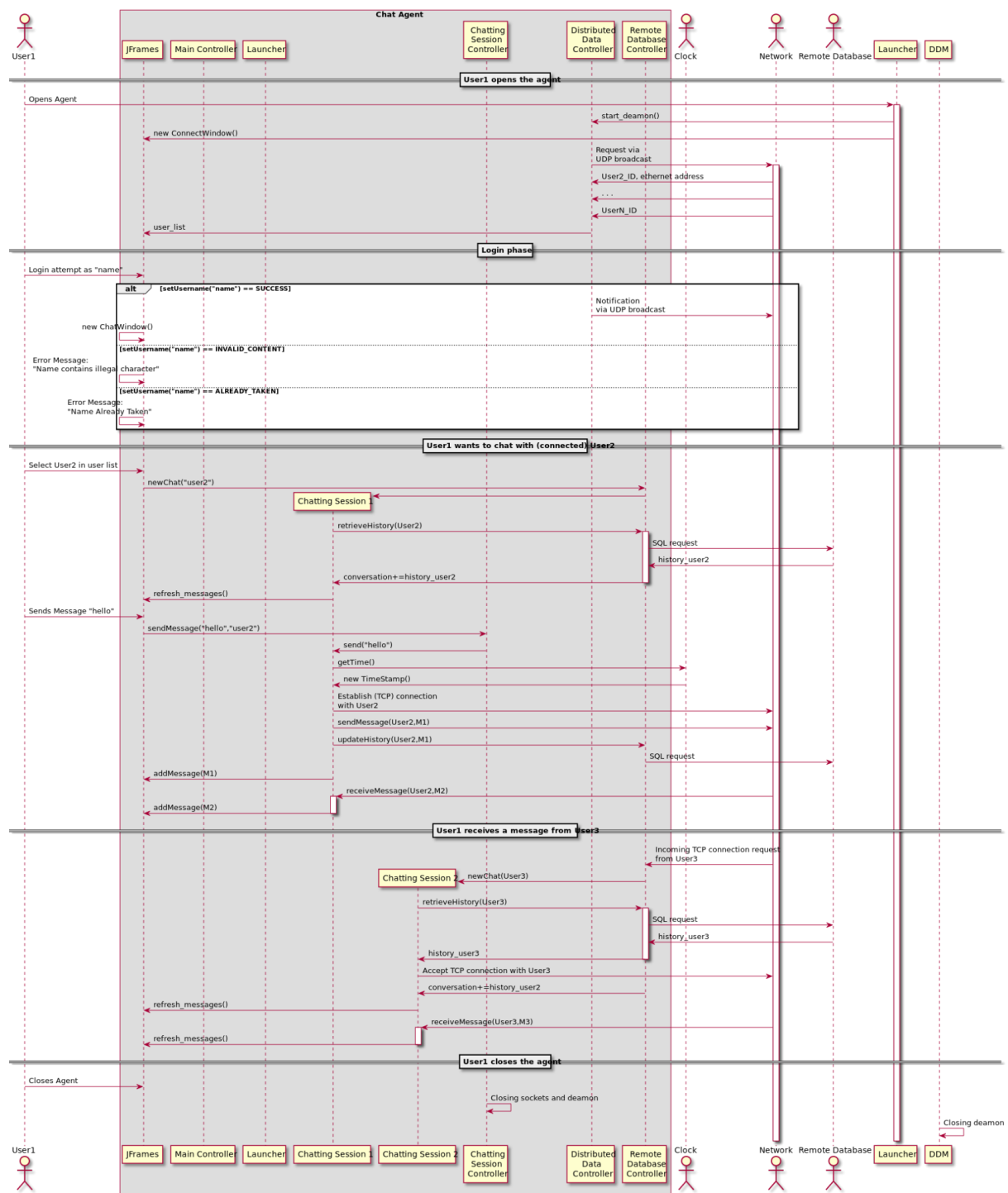
Le système, ainsi créé, va permettre, à chaque utilisateur :

- De choisir un pseudonyme unique avec lequel il sera reconnu dans ses interactions avec le système,
- D'identifier l'ensemble des utilisateurs actifs dans la liste des utilisateurs,
- De démarrer une conversation avec un utilisateur connecté (ou plusieurs utilisateurs connectés) et d'y mettre fin quand il le souhaite,
- D'informer un utilisateur connecté de la réception d'un message par un autre utilisateur connecté,
- D'avoir un historique horodaté des messages envoyées et de les afficher sur la fenêtre de chat,
- D'informer l'ensemble des utilisateurs connectés de la connexion et/ou de la déconnexion d'un utilisateur,
- D'informer l'ensemble des utilisateurs connectés du changement de pseudonyme d'un des utilisateurs

## DIAGRAMME DE SÉQUENCES



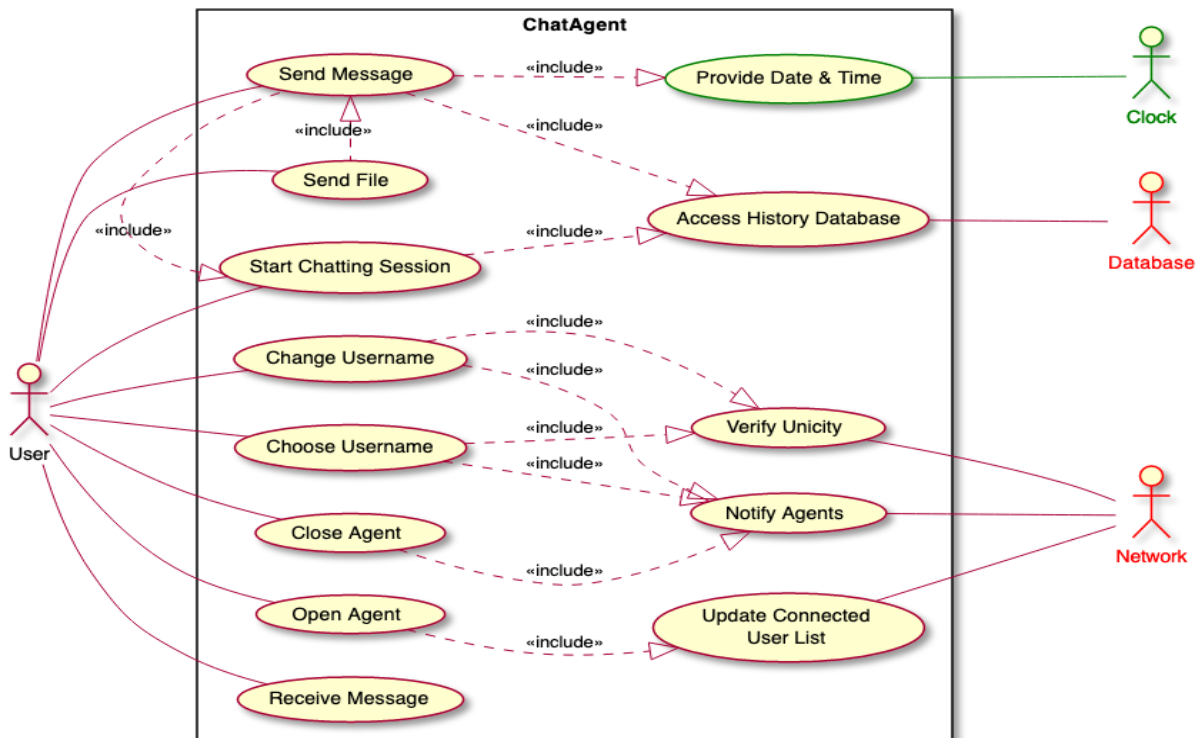
Ce diagramme permet de préciser les interactions entre l'acteur et le système de clavardage avec des messages présentés dans un ordre chronologique. Le diagramme de séquence système traite le système comme étant une boîte noire. Le comportement du système est décrit vu de l'extérieur.



Ce diagramme permet de préciser les interactions entre les acteurs et le système de clavardage avec des messages présentant encore un ordre chronologique. Le diagramme de séquence traite le système comme étant une boîte blanche. Le comportement du système est décrit vu de l'intérieur.



## DIAGRAMME DE CAS D'UTILISATION



En partant du cahier des charges et de la description qui y est faite du produit attendu, nous avons proposé une modélisation du contexte et une analyse en identifiant les principaux cas d'utilisation du système de clavardage.

## Architecture du système et choix technologiques

Les librairies utilisées sont :

SWING pour nous permettre de créer une Graphical User Interfaces (GUI) au lieu d'utiliser l'interface de console terminal, qui fournit de nombreuses classes différentes pour créer et gérer divers types d'entités d'interface utilisateur.

## MAVEN

Maven est un outil open source de Apache écrit en Java. Il est utilisé pour automatiser la gestion et la construction des projets Java.

Il peut être utilisé en ligne de commande ou via un IDE (Eclipse, IntelliJ IDEA,...).

## JDBC (Java DataBase Connectivity)

La technologie JDBC est une API permettant à notre produit de communiquer avec d'autres produits et services sans connaître les détails de leur mise en œuvre et qui fournie avec Java, la possibilité de se connecter à une base de données, constitué d'un ensemble de classes permettant de développer une application capable de se connecter à un serveur de bases de données (SGBD) de l'INSA pour ce projet.

## ● Choix techniques

### 1. Choix sur le stockage de données de conversation entres utilisateurs du Chat

Notre système de chat est une application de clavardage avec un système de stockage qui doit être décentralisé, et conformément au cahier des charges. Lors de la conception de l'application, nous avons fait le choix de ne pas déployer une base de données pour stocker les données applicatives. La solution que nous avons décidée de mettre en place pour le stockage de données est plus simple et efficace du point de vue du déploiement de l'application et de l'espace mémoire occupé par les données utilisateurs par la non-gestion d'une base de données interne.

### 2. Choix sur la signalisation entre les utilisateurs du Chat System

#### **Façon décentralisée :**

La signalisation de connexion, la déconnexion, le changement du username se fait via des paquets UDP et gérer le contrôleur Distributed\_Data pour envoyer sur le réseau par chaque utilisateur souhaitant communiquer.

La communication par messages, fichiers ou images entre les utilisateurs de l'application s'effectue toujours en utilisant la connexion en TCP.

## ● Choix conceptuels

### Choix de création et de modification du compte utilisateur

Un utilisateur qui possède un compte sur une machine est protégé grâce à un mot de passe. Si un utilisateur souhaite modifier son username, il lui faut d'abord se connecter à l'application avec son ancien login/mot de passe et modifier son login/mot de passe par la suite.

## Répartition du travail

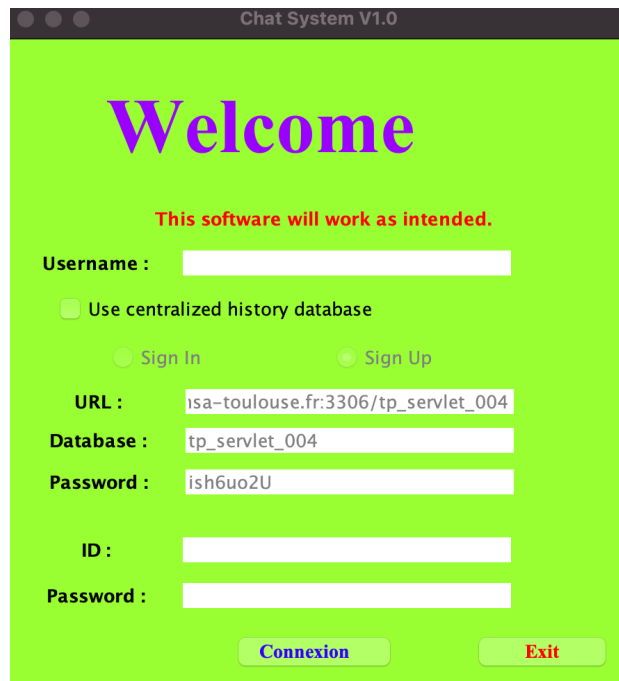
Pour ce projet, nous avons réparti le travail avec JIRA sous 6 sprints de tel façon que :

Valentin (Backend) : BDD, UDP, TCP, Contrôleur de chat, JIRA, Maven, Back End

Lauric (Frontend) : GUI, Gestion des Packages et class, Message, UserID, JIRA, Readme, Rapport



## Procédures d'évaluation et de tests



Chat System V1.0

Welcome

This software will work as intended.

Username :

☐ Use centralized history database

☐ Sign In ☒ Sign Up

URL :

Database :

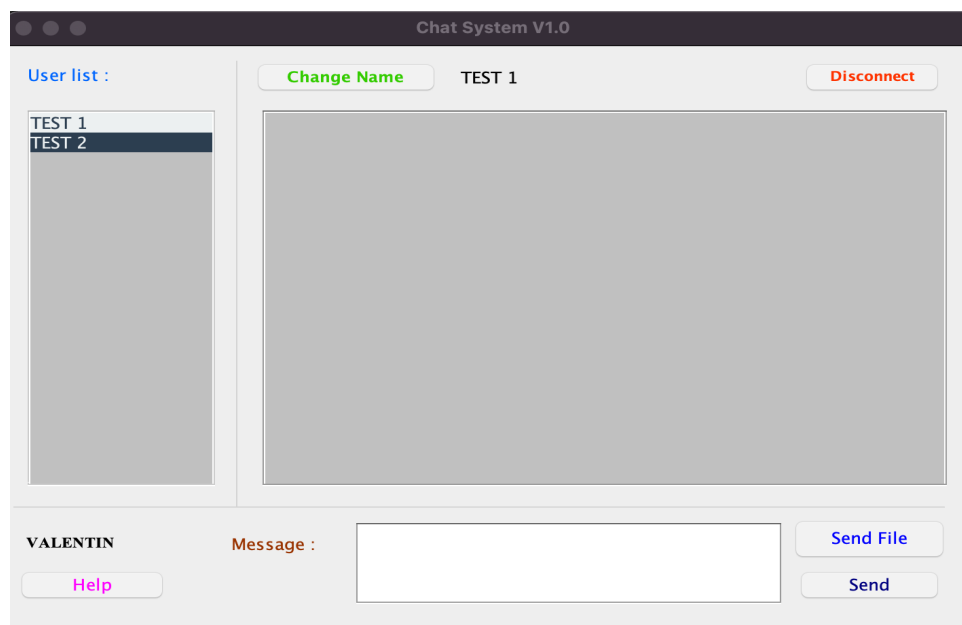
Password :

ID :

Password :

Connexion Exit

Cette fenêtre d'accueil sert à se connecter au système de tchat, grâce au bouton de connexion, elle propose une option de se connecter avec ou sans la database au autre utilisateur, il y a la possibilité de s'inscrire ou de logger à ce même serveur de stockage d'historique avec un identifiant et un mot de passe unique.



Chat System V1.0

User list :

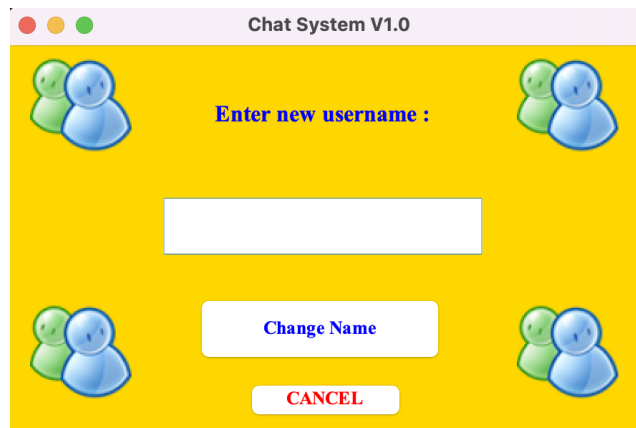
TEST 1  
TEST 2

Change Name TEST 1 Disconnect

VALENTIN Message :  Send File Send

Help

A partir de cette fenêtre, on peut discuter avec les utilisateurs une fois sélectionné dans liste utilisateur connecté et écrire du texte dans la zone Message, on y voit l'historique et les commandes "Send" et "Send File" ainsi que le bouton help et un bouton pour changer son pseudo.



Cette fenêtre sert pour le changement de nom utilisateur durant une discussion avec vérification de l'unicité de ce dernier.

## Procédure d'installation et de déploiement

Depuis un IDE  
Installez Maven  
Installation de l'application

Depuis une console terminal :  
Se rendre dans le dossier contenant le projet  
Lancer les commandes :

- ❖ mvn clean compile
- ❖ mvn package
- ❖ java -jar [fichier.jar]

## Manuel d'utilisation simplifié

Mode : FRONT END

Pour se connecter au système de chat :

Sur la fenêtre de connexion:

1. Entre votre nom d'utilisateur
2. Sélectionner la connexion à la base de donnée ou non
3. Si non appuyer sur connexion pour lancer l'agent
3. Si oui cocher si vous voulez vous inscrire "Sign up" ou s'identifier "Sign in"
4. Entrer votre ID et votre mot de passe

Ouvrir une conversation avec un autre utilisateur :

Pour envoyer un message :

1. Sélectionner un contact dans liste des utilisateurs
2. Ecrire votre message dans la zone de texte
3. Cliquer sur "Send" pour établir la connexion et envoyer le message

Pour envoyer un fichier :

Se rendre dans la fenêtre de conservation avec un autre utilisateur :

1. Sélectionner un contact dans liste des utilisateurs
2. Cliquer sur "Send file"
3. Choisir le fichier à envoyer
4. Cliquer sur ouvrir pour envoyer le fichier

Pour changer de pseudo :

1. Cliquer sur "Change name"
2. Entrer un nouveau pseudo valide
3. Cliquer sur "Change name"

Pour se déconnecter :

1. Cliquer sur "Disconnect" pour retourner à la page de connexion

Pour fermer l'application de chat :

1. Cliquer sur Exit dans la fenêtre de connexion

Si vous avez besoin de plus d'aide, appelez le service informatique au 3838.

Mode : BACK END

Il faut lancer le launcher avec un argument quelconque pour que le fichier launcher passe en mode back end.

## CONCLUSION

Ce projet est très intéressant et très prenant, il nous a permis d'acquérir de bonnes connaissances en Conception et en Programmation Orienté Objet. L'exemple pris ici est de réaliser un système de clavardage pour une entreprise qui correspond bien à notre domaine de compétence qui est l'Informatique et Réseaux.

On a pu prendre connaissance et exploiter la méthode Agile avec JIRA Software, un support simple et intuitif à utiliser pour la réalisation de notre projet.

Nous prévoyons de faire des fonctions afin d'améliorer la sécurité de l'application comme par exemple avoir la possibilité de changer de mot de passe, faire communiquer le système avec d'autres réseaux d'entreprise distant de façon sécurisée.

## BIBLIOGRAPHIE

Cahier des charges

[https://moodle.insa-toulouse.fr/pluginfile.php/26955/mod\\_resource/content/1/INSA\\_COO\\_POO\\_URD\\_v3.1.pdf](https://moodle.insa-toulouse.fr/pluginfile.php/26955/mod_resource/content/1/INSA_COO_POO_URD_v3.1.pdf)